Available online at www.sciencedirect.com

SciVerse ScienceDirect

Energy Procedia 16 (2012) 958 – 964

Energy Procedia

2012 International Conference on Future Energy, Environment, and Materials

A New Hybrid Algorithm for FPGA Routing

Yulan Tang[a] *, Jianhui Chen[b]

[1]*Wuxi City College of Vocational Technology, 214122, Wuxi, Jiangsu, China*
[2]*Wuxi Institute of Technology, 214121, Wuxi, Jiangsu, China*

Abstract

A new routing algorithm is proposed for FPGA to improve the increasing transformation cost of pseudo-Boolean Satisfiability algorithm in the routing process. The approach combines advantages of pseudo-Boolean Satisfiability and geometric routing algorithm. First, Frontier- one of geometric routing algorithm is chosen for FPGA routing. Then pseudo-Boolean Satisfiability algorithm is used when the process of Frontier is not successful. Technique of static symmetry-breaking is also adding to carry out pretreatment of pseudo-Boolean constraints, detecting and breaking the symmetries in the routing flow. The advantage is that the search path is pruned, and the cost is consequently reduced. Preliminary experiments results show that the hybrid method have no adverse affect on overall program. It has also reduced the runtime observably, and sped up the solving process.

Keywords: Benchmarking; Boolean function; Field programmable gate arrays; Routing algorithms

1 Introduction

Field-Programmable Gate Arrays (FPGAs) [1] is a programmable logic device which can implement a variety of circuit functions. It is one of the most popular devices for system prototyping, logic emulation, and so on. An FPGA consists of three portions: Configurable Logic Block (CLB), Interconnection Resource (IR) and Field-Programmable I/O Block. Because routing resource takes up 70%~80% area of the chip and about 50%~60% signal time delay [2], it is very important for FPGA to use an effect routing algorithm. It can be used to reduce the total routing area and the length of practical path wire, and improve performance of the circuit.

* Corresponding author. Tel.: 0510-85916004
E-mail address: shaoyi323@126.com.

 doi:10.1016/j.egypro.2012.01.153

All of current several geometry searching routing algorithm, such as CGE, SROUTE, PathFinder [3], VPR, Frontier [4] are based on a basic routing algorithm, named maze router, and they attempt to improve the original algorithm in different ways. Maze router finds a path between vertices on a planar rectangular grid. Each net route is started at the source logic block of the net and a search is performed through available routing resources to find the lowest cost path from source to destination. Frontier [4] is a modified maze router based on depth first search and it significantly reduces the search space required for FPGA routing and leads to decreased run time when compared to a traditional, breadth-first maze router [5]. The implementation of this algorithm is dependent on the routing sequence. A different order will get different results, which is the inherent shortcomings of the maze routing algorithm. Although Frontier has advantages of re-routing capabilities, it is can not able to determinant the routability for its dependency on routing order.

Boolean satisfiability (SAT) is a hot theme in recent years, and Pseudo-Boolean Satisfiability (PBS) [6] is a sort of special case of SAT. PBS has attracted lots of interest as a practical problem encountered in several application domains, especially in Electronic Design Automation (EDA) and Artificial Intelligence. In the EDA domain, PBS is embedded in many areas of design synthesis and validation – some of these being Automatic Test Pattern Generation (ATPG), combinatorial and sequential circuit verification, time verification, and routing. PBS is a more desirable FPGA routing approach that produces easier-to-evaluate pseudo-Boolean SAT instances than common Boolean SAT. This was possible by capturing the entire set of admissible detailed routes within a global routing region for each connection. However, there is an unavoidable limitation in this method. Because PBS is a detailed routing formulation, if a global router somehow delivers an undesirable global routing solution, there is no way for PBS to escape from this bad upper level decision. In fact, this is the major reason for PBS inferior routing solution compared to PathFinder, VPR430 and Frontier. To overcome this shortcoming, a new hybrid routing algorithm was put forward, combining the pseudo-Boolean SAT and Frontier algorithm. The new algorithm learns from other's strong points to offset one's weakness, and has achieved better performance.

The rest of the paper is organized as follows: Frontier and pseudo-Boolean SAT algorithm was briefly introduced in Sec. 2. In Sec. 3, we described the idea of new combined algorithm, and presented the routing flow of new algorithm. Static symmetry-breaking technique was also expatiated in this section. We show experimental results and analysis in Sec.4. Finally, we concluded the paper in Sec.5.

2 Preliminary Knowledge

2.1 Frontier algorithm

Frontier is developed by Dr. Tessier as his Ph.D researches in Massachuseets Institute of Technology. It is similar to timing-driven router except that it includes a domain negotiation step.

Searches between depth-first and breadth-first can be created by weighting the effect of g_i and d_i via a scaling factor α between 0 and 1:

$$f_i = (1-\alpha)\times(f_{i-1}+c_i)+\alpha\times d_i \qquad (1)$$

The node cost, c_i is used to avoid the use of nodes occupied by previous routes.

Frontier and breadth-first case is great different in the initial assignment of multi-terminal net. In the depth-first case, a specific target input must be confirmed to calculate the space d_i in equation (1). So a separate routing step will be used to ensure each input is connected with the line net. Frontier order the target input according to Prim shortest path to minimize the total wire. The target input which closest to the target output will be selected as the first input, and subsequent targets input are selected from the input

with the shortest pate from the output. In general, high fan-out is routed easier when the wiring congestion is few.

The order of searching domain is determined by lever r_d. As settled by domain negotiation, the low congested domain has lower lever r_d, so this domain will be routed at fist. The line which associated with wire resource will include r_d by modifying the formula (1).

$$f_i = (1-\alpha) \times (f_{i-1} + c_i) + \alpha \times d_i + r_d \qquad (2)$$

All the rest lines will be extended to the list by the cost function of formula (2).

2.2 Pseudo-Boolean satisfiability algorithm

In the classical Boolean SAT problem, these constraints are presented in CNF (Conjunctive Normal Form), where each constraint is a disjunction of literals. A Pseudo-Boolean (PB) constraint is a generalization of a CNF clause. A PB over a set of Boolean variable x_1, x_2, $\cdots$, x_n is an inequality of the form:

$$\sum_{i=1}^{n} c_i l_i \geq c_n + 1 \qquad (3)$$

Where $\forall i$, $c_i \in Z$ and l_i is a literal corresponding to x_i (i.e., $l_i = x_i$ or $l_i = \overline{x}_i$).

PB inequalities represent a natural generalization of CNF clauses. For example, the third CNF clause $(x_1 \vee x_2 \vee x_3)$ is equivalent to the PB constraint $x_1 + x_2 + x_3 \geq 1$. A CNF clause is a PB constraint where $\forall i$, $c_i = 1$. However, PB constraints are more expressive: a single PB constraint may in some cases correspond to an exponential number of CNF clauses [7].

A coefficient c_i is said to be activated if its corresponding literal l_i is assigned to true. A PB constraint is said to be satisfied if (1) holds. A PB formula Ψ is a conjunction of PB constraints. The problem of pseudo-Boolean satisfiability questions the existence of a truth assignment to x_1, x_2, $\cdots$, x_n satisfying all the PB constraints in Ψ. Such a truth assignment is called a satisfying assignment for Ψ.

At present, there are two types of PB solvers: first, translating the PB-SAT problem into a SAT problem and runs a state-of-the-art SAT solver [8]. This approach is particularly suited to constraint problems containing only a handful of PB constraints. Second, supporting PB constraints natively by extending the recent advancements in SAT solving to PB constraints and attempting to handle PB constraints directly. The latter is adopted in this paper.

3 New Hybrid Routing Algorithm

3.1 Hybrid routing algorithm

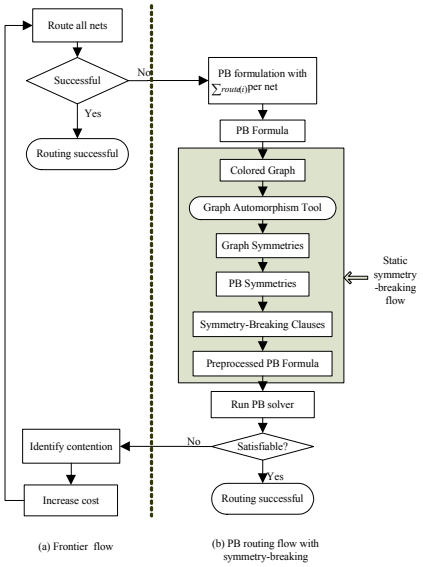
The main idea of combined algorithm is that at a certain iteration, Frontier considers only a single route for each connection. The route considered is regarded as the best route for the connection based on the latest congestion and delay cost metrics. As the iteration advances, more routing options (possibly along different global routing paths) are explored for each connection. The key idea of the combined algorithm is that at each iteration of the Frontier algorithm, we can enumerate all routing paths explored for each connection until the current iteration and examine them all simultaneously via the pseudo-Boolean SAT technique. The basic rationale behind this approach is that since only the best quality route

per two-pin connection is generated at each iteration, it is worth considering them all concurrently if no routing solution is found at some point.

The overall flow diagram of the combined algorithm is showed in Fig. 1(a), the left half is the basic Frontier flow and the right half is the PBS-based routing flow. In fact, it is not necessary to run the PBS-based flow for every iteration because of the overhead to transform a routing problem into a pseudo-Boolean SAT instance. Instead, it is most effective to run the PBS-based routing flow after a sufficient number of new detailed routes are accumulated for each signal.

Fig. 1(b) shows the pseudo code of the combined algorithm. Firstly, input is ordered by distance from node to terminal according to Prim shortest path algorithm. Then carry out domain negotiation, and adopt and extension list to save the possible wires and their corresponding costs. Pathlist is managed per two-pin connection and it has a redundancy checking ability to avoid including the exact same route per two-pin connection twice.

First, Pathlist is reset to an empty set in line 0. Line 1to 18 of the pseudo code correspond to the operation of the original Frontier algorithm. At line 20, once it is determined that there exists at least one shared routing resource, we enter the pseudo-Boolean stage. The detailed routes of each net are first decomposed into sets of two-pin connections (line 21) and inserted into the corresponding Pathlist bucket. At line 22, the route-based routing constraint formulation is performed and a routability pseudo-Boolean function Routable(X) is generated. The generated routing constraint pseudo-Boolean function Routable(X) captures all the routing constraints over a set of detailed routes explored thus far for each two-pin connection. In other words, at i-th iteration, at most i different detailed routes are considered simultaneously for each two-pin connection. At line 23, this pseudo-Boolean function is evaluated with a pseudo-Boolean SAT solver. If any routing solution is found, the algorithm returns the solution. Otherwise, the next iteration of Frontier algorithm is executed.




```
[0] Path List PL=empty
[1] Order the sinks using Prim's Algorithm.
[2] Perform Domain Negotiation.
[3] Target = sink closest to source.
[4] Put track segments attached to source onto expansion list with cost given by (2).
[5] Remove lowest cost track segment from expansion list.
[6] While the net input has not been reached.
[7]    Put neighbors of this track onto expansion list with cost given by (1).
[8]    Remove lowest cost track segment from expansion list.
[9] Endwhile
[10]Empty the expansion list.
[11]While still more sinks to route for this net.
[12]   Target = next sink determined from Prim's Algorithm.
[13]   Put whole net created to this point onto expansion list with cost = a_i.
[14]   Put track segments attached to source onto expansion list with cost given by (2).
[15]   Remove lowest cost track segment from expansion list.
[16]   While the net input has not been reached.
[17]      Put neighbors of this track onto expansion list with cost given by (1).
[18]      Remove lowest cost track segment from expansion list.
[19]   Endwhile
[20]   If there is no sink to route for this net, then return.
[21]   Convert expansion list into pathlist consist of a set of two-pin connections.
[22]   Routed-based routing constraint formulation.
[23]   Routing constraint evaluation.
[24]   If routable, then return.
[25]   Empty the pathlist.
[26]   Empty the expansion list.
[27]Endwhile
```

Fig. 1 (a) Overall flow diagram of the combined algorithm ; (b) The combined algorithm of Frontier and PBS

The new method is able to deliver a firm routability decision via pseudo-Boolean SAT technology. Once the PBS solution is found, the solution can be easily interpreted as a routing solution. If the pseudo-Boolean instance is unsatisfiable, it implies that there is no routing solution with 1) the current placement configuration and 2) the current set of detailed routes explored per two-pin connection. The information from the unroutable instance could be very valuable for a future rerouting procedure.

3.2 Symmetry-breaking flow

The technology of static symmetry-breaking for pseudo-Boolean was also used in the flow, and was showed in the shadow part of Fig. 1. It adopted the method of static symmetry-breaking to detect and break all symmetries, thus the search path was pruned, and the process of PB solver was accelerated.

The pseudo-Boolean solver was not directly used for PB inequations, and the technique of static symmetry-breaking is added to pretreat them. The shadow part of Fig. 1(a) introduces the theoretical framework for detecting and breaking symmetries in PB formulas. The basic idea is to detect all permutational symmetries using a reduction to graph automorphism, and express a PB formula as an undirected graph such that the group of the graph is isomorphic to the symmetry group of the PB formula. The presence of symmetries in the search space induces equivalence classes on the set of truth assignment. The truth assignments that satisfy equivalence classes are those assignments that set the PB formula to 1. Therefore, it is only need to consider an assignment of each equivalence calss. After adding symmetry-breaking predicates (SPBs), the lex-leaders are chosen from every class. Each symmetry is then broken by adding appropriate SBPs to the PB formula. Adding these SBPs in pretreatment is static, and an effective and un-repeated PB formula whose size is linear in the number of variables in the problem. A pseudo-Boolean solver is then applied to the preprocessed PB formula. In this paper we adopted PBS solver.

The experimental evidence in next section shows that the new routing flow significantly improves the overall routing performance which was applied to FPGA routing. Thus, the speed of solving is further accelerated.

4 Experimental Results and Analysis

In order to verify the effectiveness of this approach and facilitate the comparison, a set of standard FPGA layout benchmark circuits [9] from practical industry was used. All the three routing algorithm – Frontier, PBS, the combined algorithm of Frontier and PBS (marked as F-PBS) was applied for 15 circuit, and each circuit adopted 30 kinds of different layout. Table 1 shows results of the standard FPGA benchmarks. All experiments were run under Intel 2 GHz, 1G RAM. The operating system is Linux RedHat, and time-out is set at 1000 seconds. The pseudo-Boolean solver PBS is used here. The first column of Table 1 lists the benchmarks. The naming convention encodes the formulation style (“gr” for global routing before detailed routing; “res” or “2pin” for formulation), and the number of tracks per channel in the FPGA (e.g., “w7” means 7 tracks). Columns 3, 4, 5 are show the size of FPGA benchmarks. And the overall computing time for three algorithms is respectively shown in columns 6, 7, 8. The best results are denoted in bold.

According to table 1, the new hybrid algorithm combines the advantages of Frontier and PBS, compared to the pure geometric routing algorithm, Frontier. It also applied the static symmetry-breaking technology to reduce the search path, significantly speeds up the solving progress. Because the routing method based on PBS belongs to a sort of parallel methods, it is able to route multiple wire at one time. In other words, different form traditional one-net-at-a-time approach, this algorithm has nothing to do with the order of connection. It is very important that in the case of a given layout, PBS can quickly predicate routability. It is effectively complement the deficiency of Frontier in this area.

5 Conclusions

Routing for FPGAs is a very challenging problem due to the limitation of routing resources. Although in a given layout, PBS-based technique could determinate routability of the circuit soon, the cost of transforming to PBS is high. The pure PBS-based method still has a fundamental limitation that the scalable of the problem it produced is much smaller than general routing methods. However, Frontier of geometric routing method has no such restrictions. But Frontier does not have the ability to predict the routability. So the combined algorithms (PBS and Frontier) complement each other. The new routing flow also uses static symmetry-breaking technology in the pretreatment stage to deal with pseudo-Boolean constraints. Experimental results show that the new hybrid routing algorithm observably reduce the runtime. The development of pseudo-Boolean SAT algorithm and technology of symmetry-breaking will enhance the ability of solving routing problems of FPGAs.

Table 1. Comparison of running time for FPGA routing instances using three routing algorithms

FPGA Benchmark		Instance size		Time(s)		
Name	Nets	Variables	Clauses	Frontier	PBS	F-PBS
9symml_gr_2pin_w_w5.cnf	79	2604	32450	120.18	59.99	54.76
9symml_gr_rcs_w5.cnf	79	1295	24309	65.34	11.74	11.4
alu2_gr_2pin_w7.cnf	153	3882	84209	40.21	26.52	21.33
alu2_gr_rcs_w7.cnf	153	3570	73478	38.65	18.96	10.46
apex7_gr_2pin_w4.cnf	126	1322	10940	120.18	9.95	4.89
apex7_gr_res_w4.cnf	126	1200	9416	87.18	7.66	4.76
C499_gr_2pin_w5.cnf	115	2070	19908	71.78	460.76	70.8
C499_gr_rcs_w5.cnf	115	1560	15777	67.83	343	53.32
C880_gr_2pin_w6.cnf*	234	4623	62711	>1000	>1000	>1000
C880_gr_rcs_w6.cnf*	234	3936	53018	444.52	472	352.6
term1_gr_2pin_w3.cnf	88	746	3517	28.72	267	27.66
term1_gr_rcs_w3.cnf	88	606	2518	13.97	11.4	9.46
too_large_gr_2pin_w6.cnf*	186	3972	52678	>1000	894	445.7
too_large_gr_rcs_w6.cnf*	186	3114	43251	900.70	710	583.44
example2_gr_2pin_w5.cnf	205	3603	36334	93.38	>1000	93.32

Acknowledgements

It is a project supported by college-level issue of Wuxi City College of Vocational Technology (The Study of the Particle Swarm Optimization and Its Application in PID controller) (WXCY-2011-GY-008).

References

[1] Gudise, V., Venayagamoorthy, G.: FPGA Placement and Routing Using Particle Swarm Optimization. in Proceedings of the IEEE Computer Society Annual Symposium on VLSI Emerging Trends in VLSI Systems Design, vol: 12, pp. 307-308 (2004).

[2] Mo, F., Tabbara, A., Brayton, R.: A Force-Directed Maze Router, Department of EECS. University of California at Berkeley. 2001,p. 404-407.

[3] McMurchie, L., Ebeling, C.: PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs. in ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, ACM Press, New York NY, 1995, p: 111-117.

[4] Tessier, R.: Negotiated A* Routing for FPGAs. in Proceedings of the Fifth Canadian Workshop on Field-Programmable Devices, Montreal, Quebec, Canada, 1998, p. 14-19.

[5] Hwang, C. H., Allen, C.: A Predictive System Shutdown Method for Energy Saving of Event-Driven Computation. ACM Transactions on Design Automation of Electronic Systems, vol:5, 2000, p. 226-241.

[6] Sheini, H. M., Sakallah, K. A.: Pueblo: A Hybrid Pseudo-Boolean SAT Solver. Boolean Modeling and Computation, 2006, p.155-179.

[7] Aloul, F. A., Ramani, A.: ShatterPB: Symmetry-Breaking for Pseudo-Boolean Formulas. IEEE, 2004, p. 884-887.

[8] Niklas, E., Niklas, S.: Translating pseudo-Boolean constraints into SAT. Journal on Satisfiability, Boolean Modeling and Computation,2006.

[9] SEGA Detailed Routing Software, http://www.eecg.toronto.edu/~lemieux/sega/sega.html